

Inicio | Fundamentos | Instalación | Añadir Elementos | Encadenar Métodos | Asociar Datos | Usando sus Datos | Desplegar DIVs | La Función `data()` | Introducción a SVG | Despliegue de SVG | Tipos de Datos | Diagrama de Barras | Diagrama de Dispersión | Escalas | Ejes |

Asociar Datos

En qué consiste Asociar Datos (Binding) y por qué es aconsejable?

La visualización de datos es un proceso de *mapeo* de datos a despliegues gráficos. Los datos ingresan y su representación en formato gráfico constituyen el resultado final. De esa manera es probable que números mayores correspondan a barras más altas o que categorías especiales correspondan a colores más llamativos. Las reglas de mapeo son decisión del programador.

Con D3, se *asocian* los datos de entrada a los elementos del DOM. *Asociar* en este sentido es similar a *pegarse a* elementos específicos, de tal forma que posteriormente se puedan referenciar esos valores y aplicar la reglas de mapeo. Si no se lleva a cabo el paso de asociación, lo único que se obtiene es un listado de elementos del DOM sin datos que no pueden ser mapeados. A nadie le interesa eso.

En Asocio

En D3 se utiliza el método **selection.data()** para asociar datos a los elementos del DOM. Sin embargo, es necesario tener dos componentes organizados antes de poder asociar datos:

1. Los datos
2. Una selección de elementos del DOM

Miremos uno a la vez.

Datos

D3 es bastante inteligente con respecto al manejo de diferentes tipos de datos y acepta casi cualquier tipo de arreglo de números, caracteres u objetos (éstos a su vez pueden contener otros arreglos o pares de llave/valor). D3 puede leer JSON (y GeoJSON) con facilidad y tiene también un método que ayuda a montar archivos en formato CSV.

Con el propósito de mantener la simplicidad, por el momento vamos a empezar con un arreglo simple de números. Esta es la muestra:

```
var dataset = [ 5, 10, 15, 20, 25];
```

Por Favor Haga su Selección

Primero es indispensable decidir qué es lo que se va a seleccionar, o sea, qué elementos estarán asociados a los datos? Nuevamente, se utilizará un esquema sencillo - esta vez se va a crear un párrafo nuevo para cada valor en el conjunto de datos. Es posible imaginarse que algo así puede ser útil:

```
d3.select("body").selectAll("p")
```

Es correcto, sin embargo toca tener algo en cuenta: Los párrafos que queremos seleccionar *aún no existen*. Este es uno de los puntos que más genera confusión con respecto a D3. Cómo puede ser posible seleccionar elementos que aún no existen? Paciencia por favor - la respuesta puede requerir la maleabilidad de su mente.

La respuesta está en el método `enter()`, que es verdaderamente mágico. Acá está el código final del ejemplo, que será explicado a continuación:

```
d3.select("body").selectAll("p")  
  .data(dataset)  
  .enter()  
  .append("p")  
  .text("New paragraph!");
```

Ahora, revise lo que hace el código con la página de ejemplo. Puede ver cinco párrafos nuevos, cada uno con el mismo contenido. Esto es lo que pasa:

`d3.select("body")` - encuentra la etiqueta `-body-` en el DOM y pasa la referencia al siguiente paso en la cadena.

`.selectAll("p")` - Selecciona todos los párrafos del DOM. Como aún no existen, devuelve una selección vacía. Esto se debe entender como la representación de los párrafos que existirán *próximamente*.

`.data(dataset)`. Cuenta y desagrega los valores. Existen cinco valores en la muestra, por consiguiente de aquí en adelante todo se ejecutará cinco veces, una vez por cada valor.

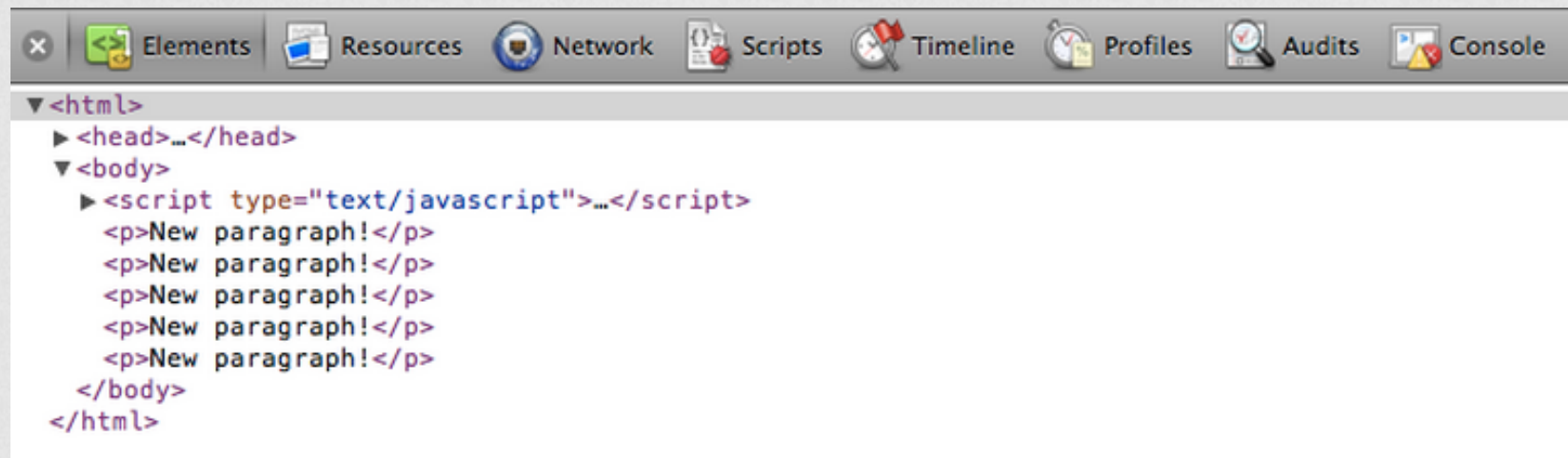
`.enter()` - Para crear nuevos elementos que estén asociados a datos, es necesario utilizar `enter()`. Este método revisa el DOM y luego revisa los datos que le serán entregados. Si existen más datos que elementos del DOM, `enter()` *crea un elemento temporal* con el cual se puede hacer la magia. Luego `enter()` entrega la referencia al elemento temporal al siguiente paso en la cadena.

`.append()` - Utiliza la selección temporal creada previamente por el método `enter()` e inserta el elemento `p` en el DOM. Qué bien! Ahora lo pasa como referencia al elemento que acaba de crear al siguiente paso en la cadena.

`.text("New paragraph!")` - Toma la referencia al recién creado **p** e inserta el valor del texto.

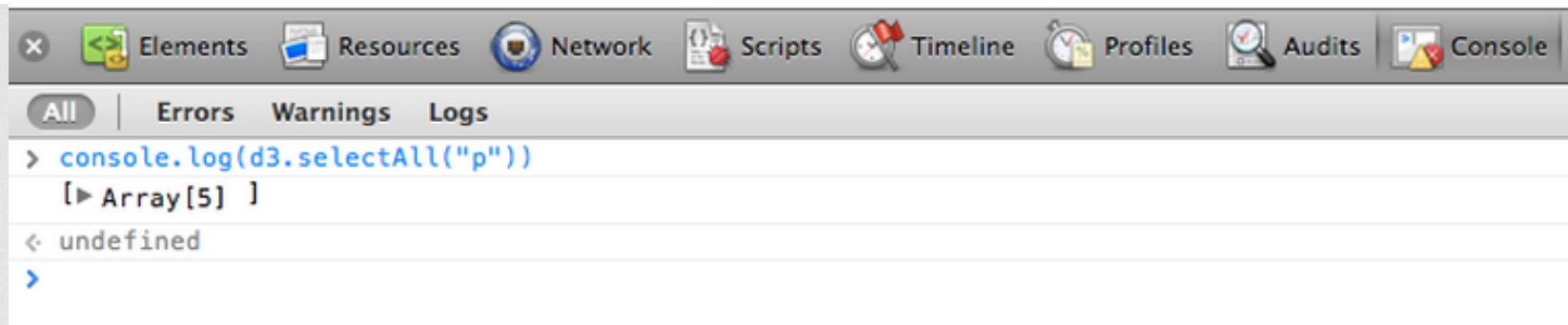
Asociados y Listos

Muy bien! Los datos se pudieron leer, desagregar y asociar a los nuevos elementos **p** que se crearon en el DOM. Si no cree, puede regresar a la [página de ejemplo](#) y abrir el inspector de web.

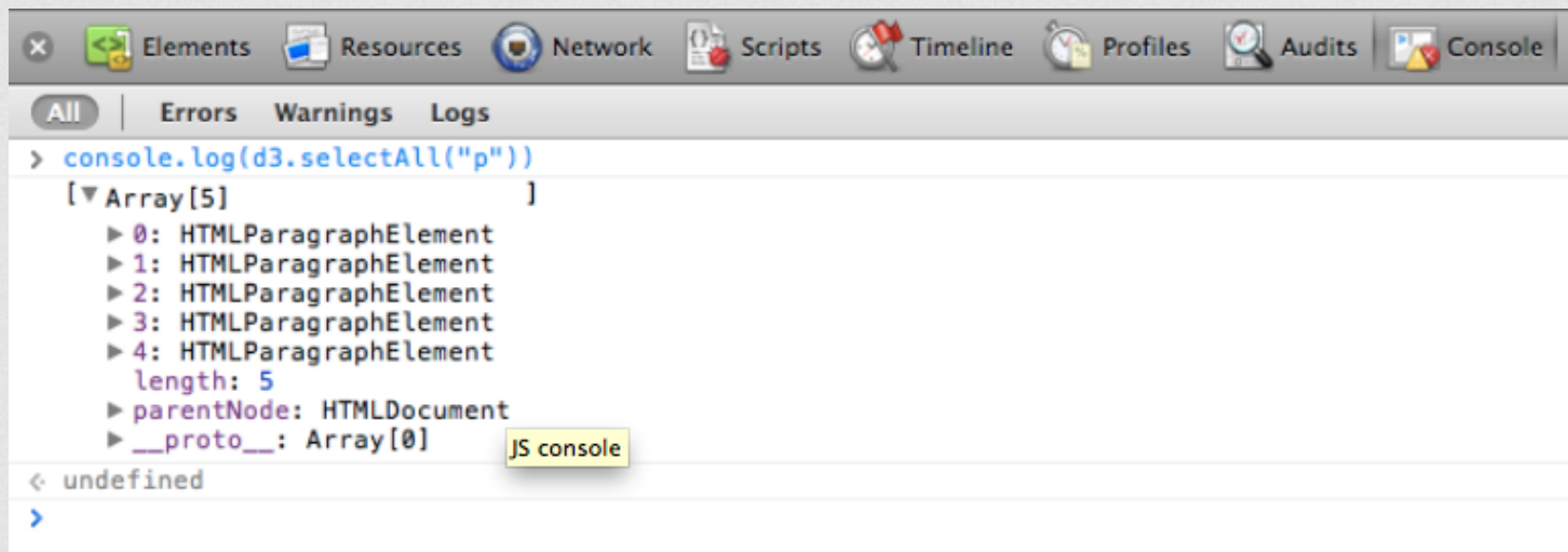


Ya se tienen los cinco párrafos, pero donde están los datos? Oprima la opción *Consola*, y escriba el siguiente código de JavaScript y luego oprima la tecla Enter:

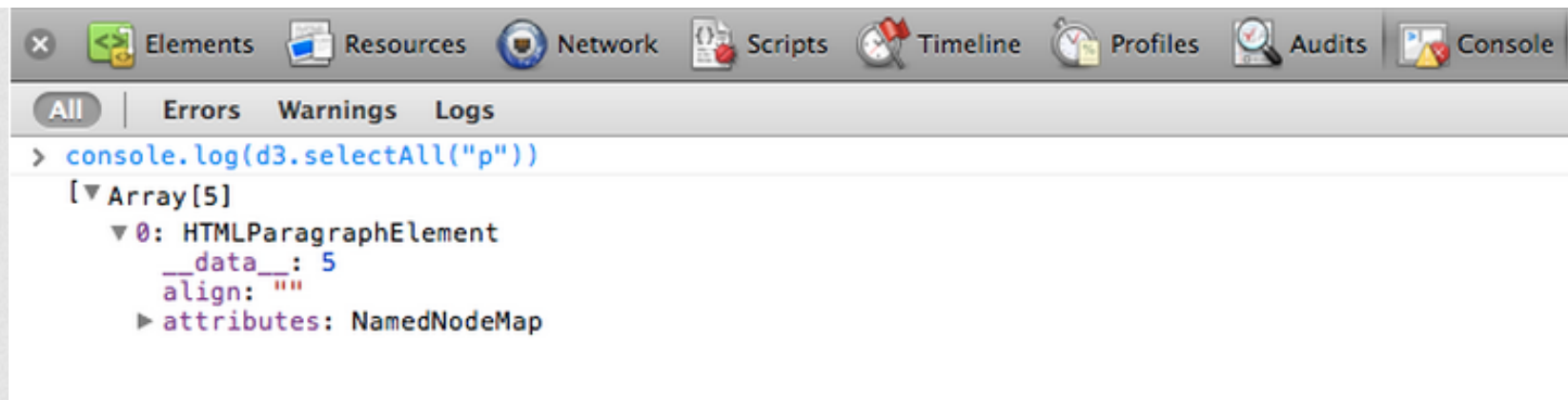
```
console.log(d3.selectAll("p"))
```



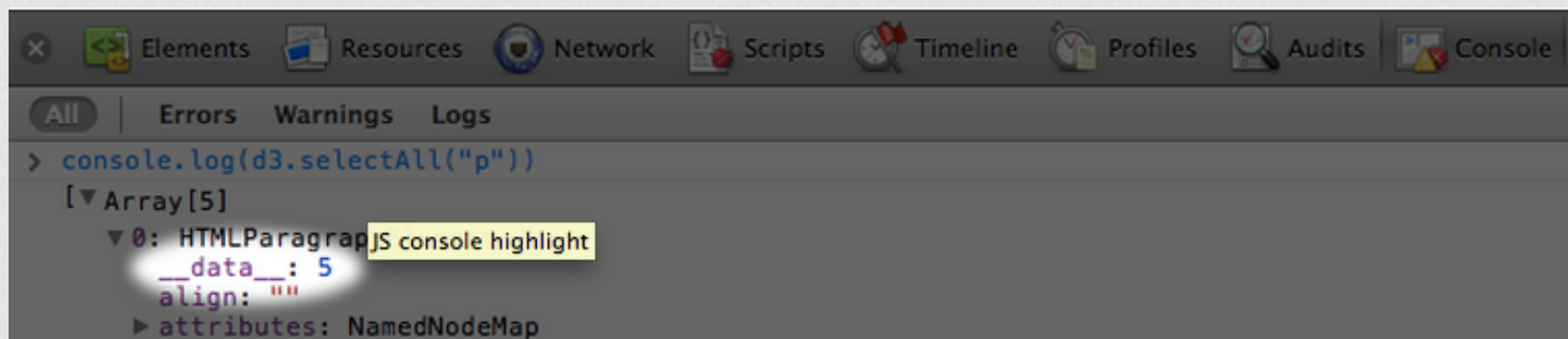
Un arreglo! Ahora oprima el triángulo gris para ver más:



Ahora podrá ver los cinco **HTMLParagraphElements**, numerados del 0 al 4. Oprima el triángulo para ver el primero (número 0)



Lo ve? Realmente lo ve? Increíble. Ahí está:



El primer valor, el número 5, se puede ver bajo el atributo `__data__` del párrafo. Si oprime los demás elementos de párrafo, verá que también contienen valores en el atributo `__data__` 10, 15, 20 y 25, tal como se especificaron.

Los datos están listos. Es hora de hacer algo con ellos.

Siguiente: Usando sus Datos →

Este tutorial fue traducido por Gabriel Coch

Todo el contenido fue desarrollado por y le pertenece a Scott Murray